[redacted]

UIUCDCS-R-72-498

A NUMERICAL SOLUTION OF THE MATRIX RICCATI EQUATIONS

By

Killion Noh

January 20, 1972

CAC Document No. 24

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

CAC Document No. 24

UIUCDCS-R-72-498

A NUMERICAL SOLUTION OF THE MATRIX RICCATI EQUATIONS

By

Killion Noh

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

January 20, 1972

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign and supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the U.S. Army Research Office-Durham under Contract No. DAHC04 72-C-0001.

ABSTRACT

The eigenvector solution of the time-invariant matrix Riccati equation is discussed. The coefficient matrix of the canonical equation is allowed to have multiple eigenvalues, namely, the matrix could be either derogatory or defective. The solution of matrix Riccati equation is then calculated from a part of similarity transformation which should reduce the coefficient matrix to the Jordan canonical form.

ACKNOWLEDGEMENT

The author would like to express his thanks to his advisor, Professor Daniel L. Slotnick, and to the ILLIAC IV Project and the Center for Advanced Computation under whose employ this research was conducted. A note of appreciation is also extended to the Department of Computer Science for its cooperation and support during the time of the author's studies at the University of Illinois.

Deepest gratitude also goes to Professor Ahmed H. Sameh for suggesting this thesis topic and for his guidance throughout its preparation. Without his encouragement and patience, this work would have never been done.

TABLE OF CONTENTS

1. INTRODUCTION

It is well known [1,2] that the linear regulator problem with quadratic cost functional is reduced to the problem of the matrix Riccati equation.

Although the solution of the matrix Riccati equation exists and is unique [2,3], it is not easy to obtain a numerical solution for the high-order system.

One of the numerical methods is the eigenvector solution which has been studied by several authors [4,5,6]. In their studies, however, the coefficient matrix of the canonical equation is assumed to have distinct eigenvalues, which is a restriction to the method of eigenvector solutions. Recently, Martensson [7] discussed the case in which the coefficient matrix is nondiagonable. Similarly, in this paper, the assumption of distinct eigenvalues will also be removed so that the coefficient matrix is allowed to have the multiple eigenvalues, namely, the matrix is either derogatory or defective. To do this, we explicitly construct the similarity transformation which should reduce the coefficient matrix to the Jordan canonical form, either diagonal or nondiagonal, and then the solution of the matrix Riccati equation is calculated from a part of the similarity transformation matrix.

Since matrix Riccati equations of fairly large sizes appear in many engineering applications and since the solution algorithm can be easily constructed such that it becomes suitable for a parallel computer [8], we provide in Appendix II the basic codes written in ILLIAC IV language, GLYPNIR.

2. FORMULATION OF THE PROBLEM

In this section, the source of the matrix Riccati equation and its relation to optimal control problems are summarized in short for our further discussions [1,2,3].

Let us consider the linear time-invariant dynamical system

$$\begin{aligned} x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \\ x(t_0) &= x_0 \end{aligned} \tag{2.1}$$

where A, B, and C are n x n, n x r, and m x n constant matrices, respectively, and x(t) is an n-dimensional state vector, u(t) is an r-dimensional control vector, and y(t) is an m-dimensional output vector, respectively. Further, we assume that the system (2.1) is completely observable. The optimal output regulator problem is then to determine the control u(t) which minimizes the quadratic cost functional,

$$V_1 = \frac{1}{2} y^T (t_f) \quad Py(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (y^T(t) \; Qy(t) + u^T(t)Ru(t))dt \tag{2.2}$$

where P and Q are m x m symmetric positive semidefinite matrices and R is an r x r symmetric positive definite matrix. Substituting y(t) = Cx(t) into (2.2), V_1 is rewritten as

$$V_1 = \frac{1}{2} x^T(t_f) \; C^TPCx(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x^T(t)C^TQC \; x(t) + u^T(t)Ru(t))dt \tag{2.3}$$

Since the system (2.1) is completely observable, C^TPC and C^TQC are symmetric and positive semidefinite [1].

The optimal feedback control, therefore, is given by

$$u^*(t) = -R^{-1}B^TK(t)x(t) \tag{2.4}$$

where K(t) is an n x n symmetric positive definite matrix which is the solution of the matrix Riccati equation

$$\dot{K}(t) + K(t)\,A + A^T K(t) - K(t)BR^{-1}B^T K(t) + C^T QC = 0 \qquad (2.5)$$

with the boundary condition

$$K(t_f) = C^T PC \qquad (2.6)$$

Furthermore, the optimal trajectory is the solution of the system of differential equations

$$\dot{x}(t) = (A - BR^{-1}B^T K(t))\, x(t) \qquad (2.7)$$

$$x(t_0) = x_0$$

The minimum cost is given by

$$V_1{}^* = \frac{1}{2}\, x^T(t)\; K(t)\; x(t) \qquad (2.8)$$

In addition to the assumption of complete observability we further assume that the system is completely controllable. Setting P = 0 and $t_f = \infty$, the output regulator problem turns out to be the following:

minimize

$$V = \frac{1}{2} \int_0^\infty (y^T(t)Qy(t) + u^T(t)Ru(t))\, dt \qquad (2.9)$$

subject to

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) \qquad (2.10)$$

$$x(0) = x_0$$

Then the optimal control is given by

$$u(t) = -R^{-1}B^T K\, x(t) \qquad (2.11)$$

where K is the n x n symmetric positive definite matrix which is the solution of algebraic Riccati equation

$$KA + A^T K - KBR^{-1}B^T K + C^T QC = 0 \qquad (2.12)$$

The optimal trajectory is the solution of the system

$$\begin{aligned} \dot{x}(t) &= Gx(t) \\ G &= A - BR^{-1}B^TK \\ x(0) &= x_0 \end{aligned} \tag{2.13}$$

Note that for an optimal control system the matrix G is diagonable and has eigenvalues with negative real parts. Therefore, the symmetric positive definite solution K of (2.12) must be such that G has eigenvalues with negative real parts.

Kalman [2] has shown that if the system (2.1) is completely observable and completely controllable, then

$$\lim_{t_f \to \infty} K(t_f) = K \tag{2.14}$$

We shall observe this equation in the following.

Let $K(t) = Z(t)X^{-1}(t)$ (2.15)

where X(t) and Z(t) are n x n matrices and X(t) is non-singular. Substituting

$$\dot{K}(t) = (\dot{Z}(t) - K(t)\ \dot{X}(t))\ X^{-1}(t) \tag{2.16}$$

into (2.5) and setting

$$\dot{X}(t) = AX(t) - BR^{-1}B^T\ Z(t) \tag{2.17}$$

we obtain

$$\dot{Z}(t) = -C^TQCX(t) - A^TZ(t) \tag{2.18}$$

Therefore (2.5) becomes a system of 2n-dimensional linear homogeneous differential equations

$$\begin{bmatrix} \dot{X}(t) \\ \dot{Z}(t) \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -C^TQC & -A^T \end{bmatrix} \begin{bmatrix} X(t) \\ Z(t) \end{bmatrix} \tag{2.19}$$

Since we are interested in increasing the time t, let

$$\tau = t_f - t \tag{2.20}$$

Then

$$\begin{bmatrix} \dot{X}(\tau) \\ \dot{Z}(\tau) \end{bmatrix} = W \begin{bmatrix} X(\tau) \\ Z(\tau) \end{bmatrix} \qquad (2.21)$$

where

$$W = \begin{bmatrix} -A & BR^{-1}B^T \\ C^TQC & A^T \end{bmatrix}$$

with

$$X(0) = I$$

$$Z(0) = C^TQC \qquad (2.22)$$

Therefore, the solution will be given by the following matrix exponential:

$$\begin{bmatrix} X(\tau) \\ Z(\tau) \end{bmatrix} = e^{W\tau} \begin{bmatrix} I \\ C^TQC \end{bmatrix} \qquad (2.23)$$

Consequently, the solution of the matrix Riccati equation will be obtained by (2.15) as $\tau \rightarrow \infty$. In the next section, we shall show that two matrices $X(\tau)$ and $Z(\tau)$ are obtained from the similarity transformation which reduces W into the Jordan canonical form. Throughout the remaining discussions, the increasing time variable t will be used instead of τ.

3. SOLUTION OF THE MATRIX RICCATI EQUATION

We begin with two lemmas for our main result.

Lemma 1. The eigenvalues of the matrix W of (2.21) are symmetric with respect to the imaginary axis.

Proof. Let I_1 be a 2n x 2n matrix given by

$$I_1 = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \tag{3.1}$$

where I is an n x n identity matrix. Then it is obvious that

$$I_1^{-1} = I_1^T \tag{3.2}$$

$$I_1 W I_1^T = -W^T \tag{3.3}$$

Since the eigenvalues of a matrix are unchanged by either similarity transformations or transposing, the eigenvalues of W occur in pairs with opposite signs.

Lemma 2. Let

$$T = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{3.4}$$

where A_{11}, A_{22}, A_{12}, and A_{21} are n x n, m x m, n x m, and m x n matrices, respectively. Assume that A_{11} is non-singular. Then

$$T^{-1} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \tag{3.5}$$

exists iff $A_{22} - A_{21} A_{11}^{-1} A_{12}$ is non-singular, and is given by

$$\begin{aligned} T_{11} &= A_{11}^{-1} + A_{11}^{-1} A_{12} T_{22} A_{21} A_{11}^{-1} \\ T_{12} &= -A_{11}^{-1} A_{12} T_{22} \\ T_{21} &= -T_{22} A_{21} A_{11}^{-1} \\ T_{22} &= (A_{22} - A_{21} A_{11}^{-1} A_{12})^{-1} \end{aligned} \tag{3.6}$$

where submatrices T_{11}, T_{22}, T_{12}, and T_{21} are the same sizes as A_{11}, A_{22}, A_{12}, and A_{21}, respectively.

Proof. Suppose $A_{22} - A_{21} A_{11}^{-1} A_{12}$ is non-singular. From $TT^{-1} = I$,

$$\begin{aligned} A_{11} T_{11} + A_{12} T_{21} &= I_n \\ A_{11} T_{12} + A_{12} T_{22} &= 0 \\ A_{21} T_{11} + A_{22} T_{21} &= 0 \\ A_{21} T_{12} + A_{22} T_{22} &= Im \end{aligned} \tag{3.7}$$

Premultiplying the second equation by $A_{21} A_{11}^{-1}$ and subtracting from the fourth,

$$T_{22} = (A_{22} - A_{21} A_{11}^{-1} A_{12})^{-1}$$

And hence

$$T_{12} = -A_{11}^{-1} A_{12} T_{22}$$

Similarly, from the first and third,

$$T_{21} = -T_{22} A_{21} A_{11}^{-1}$$

$$T_{11} = A_{11}^{-1} + A_{11}^{-1} A_{12} T_{22} A_{21} A_{11}^{-1}$$

Hence T^{-1} is a right inverse of T. In a similar way we can show that T^{-1} is also a left inverse of T. Conversely, suppose T is non-singular. Then

$$0 \neq \det \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \det \left(\begin{bmatrix} A_{11} & 0 \\ A_{21} & I_m \end{bmatrix} \begin{bmatrix} I_n & A_{11}^{-1} A_{12} \\ 0 & A_{22} - A_{21} A_{11}^{-1} A_{12} \end{bmatrix} \right)$$

$$= \det (A_{11}) \det(A_{22} - A_{21} A_{11}^{-1} A_{12}) \tag{3.8}$$

Therefore, $A_{22} - A_{21} A_{11}^{-1} A_{12}$ is nonsingular.

Assuming that W has distinct eigenvalues, we now turn to constructing the similarity transformation with which W of (2.21) is reduced to the diagonal form. By lemma 1, W has n eigenvalues with positive real parts and n eigenvalues with negative real parts. If we construct the similarity transformation S, whose columns are the right eigenvectors of W, such that the first n columns are the right eigenvectors of W corresponding to the n eigenvalues of W with positive real parts, then

$$S^{-1}WS = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_{2n} \end{bmatrix} \qquad (3.9)$$

where, of course, the rows of S^{-1} are the left eigenvectors of W. O'donnell [5] and Potter [6] have shown that if we partition S into four n x n submatrices,

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \qquad (3.10)$$

then the solution of the matrix Riccati equation is given by

$$K = S_{21} \, S_{11}{}^{-1} \qquad (3.11)$$

provided that S_{11} is non-singular.

We shall now remove the restriction that W has distinct eigenvalues and present an alternative proof to that of Martensson [7] to show that (3.11) holds for the case in which W has nondiagonal Jordan form.

Let us consider the general Jordan form J with the similarity transformation S, which will be constructed later,

$$S^{-1}WS = J$$

where, $$J = \begin{bmatrix} J_{m_1} & & \\ & \ddots & \\ & & J_{m_\ell} \end{bmatrix} \qquad (3.12)$$

and in which

$$J_{m_i} = \begin{bmatrix} \lambda_i & 1 & & & \\ & \lambda_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda_i \end{bmatrix}, \quad (m_i \times m_i) \tag{3.13}$$

If we denote S by its column vectors $s_1, \ldots, s_{2n}$, then

$$W\,[s_1, s_2, \ldots, s_{2n}] = [s_1, s_2, \ldots, s_{2n}]\,J \tag{3.14}$$

For any Jordan block J_{m_i}, if $m_i > 1$ then, dropping subscripts of s for convenience,

$$\begin{aligned} Ws^{(1)} &= s^{(1)}\lambda_i \\ Ws^{(2)} &= s^{(1)} + \lambda_i\, s^{(2)} \\ &\cdots\cdots\cdots \\ Ws^{(m_i)} &= s^{(m_i-1)} + \lambda_i\, s^{(m_i)} \end{aligned} \tag{3.15}$$

or

$$\begin{aligned} (W - \lambda_i I)\, s^{(1)} &= 0 \\ (W - \lambda_i I)^2\, s^{(2)} &= 0 \\ &\cdots\cdots\cdots \\ (W - \lambda_i I)^{m_i}\, s^{(m_i)} &= 0 \end{aligned} \tag{3.16}$$

Thus, $s^{(j)}$, $1 \leq j \leq m_i$, is a principal vector of degree j associated with λ_i. Since S is nonsingular, the m_i principal vectors $s^{(1)}, \ldots, s^{(m_i)}$ are independent. If we consider all the Jordan blocks J_{m_i}, $1 \leq i \leq \ell$, we have the 2n principal vectors $s_1, s_2, \ldots, s_{2n}$ which constitute the similarity transformation S in (3.12). Further, we rearrange the columns of S such that the first n vectors are the principal vectors corresponding to the n eigenvalues with positive real parts of W.

Let us now split J of (3.12) into

$$J = \left[\begin{array}{c:c} J_1 & \\ \hdashline & J_2 \end{array}\right] + \left[\begin{array}{c:c} E_1 & \\ \hdashline & E_2 \end{array}\right] \tag{3.17}$$

where J_1 and J_2 are diagonal with the eigenvalues of W with positive and negative real parts respectively, and E_1 and E_2 are matrices with one's on the super-diagonals and zeros elsewhere. To associate with (3.6), let us partition S and S^{-1} as

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$

$$S^{-1} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \tag{3.18}$$

where all submatrices are n x n. Note that, by lemma 2, it is necessary to assume that S_{11} and $S_{22} - S_{21}\, S_{11}^{-1}\, S_{12}$ are nonsingular.

Then

$$e^{Wt} = e^{SJS^{-1}t}$$

$$= Se^{Jt}S^{-1}$$

$$= \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} e^{J_1t} & 0 \\ 0 & e^{J_2t} \end{bmatrix} \begin{bmatrix} e^{E_1t} & 0 \\ 0 & e^{E_2t} \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}$$

$$= \begin{bmatrix} S_{11}e^{J_1t}e^{E_1t}T_{11} + S_{12}e^{J_2t}e^{E_2t}T_{21} & S_{11}e^{J_1t}e^{E_1t}T_{12} + S_{12}e^{J_2t}e^{E_2t}T_{22} \\ S_{21}e^{J_1t}e^{E_1t}T_{11} + S_{22}e^{J_2t}e^{E_2t}T_{21} & S_{21}e^{J_1t}e^{E_1t}T_{12} + S_{22}e^{J_2t}e^{E_2t}T_{22} \end{bmatrix} \tag{3.19}$$

Since the elements of J_2 have negative real parts, $e^{J_2 t} \to 0$ as $t \to \infty$ and thus the terms having $e^{J_2 t}$ in (3.19) vanish as $t \to \infty$. Therefore,

$$\begin{bmatrix} X(t) \\ Z(t) \end{bmatrix} = e^{Wt} \begin{bmatrix} I \\ C^TQC \end{bmatrix}$$

$$= \begin{bmatrix} S_{11}e^{J_1 t}e^{E_1 t}T_{11} & S_{11}e^{J_1 t}e^{E_1 t}T_{12} \\ S_{21}e^{J_1 t}e^{E_1 t}T_{11} & S_{21}e^{J_1 t}e^{E_1 t}T_{12} \end{bmatrix} \begin{bmatrix} I \\ C^TQC \end{bmatrix}$$

$$= \begin{bmatrix} S_{11}\ e^{J_1 t}\ e^{E_1 t}\ (T_{11} + T_{12}\ C^TQC) \\ S_{21}\ e^{J_1 t}\ e^{E_1 t}\ (T_{11} + T_{12}\ C^TQC) \end{bmatrix} \qquad (3.20)$$

Therefore the solution of the matrix Riccati equation is given by

$$K = \lim_{t\to\infty}\ Z(t)\ X^{-1}(t)$$

$$= \lim_{t\to\infty} [S_{21}\ e^{J_1 t}\ e^{E_1 t}\ (T_{11}+T_{12}\ C^TQC)][S_{11}e^{J_1 t}e^{E_1 t}(T_{11}+T_{12}C^TQC)]^{-1}$$

$$= S_{21}\ S_{11}^{-1} \qquad (3.21)$$

provided that the indicated inverse exists. However, we shall show that only the nonsingularity of S_{11} and $S_{22} - S_{21}\ S_{11}^{-1}\ S_{12}$ is needed for the stationary solution of the matrix Riccati equation. From (3.12),

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} -A & BR^{-1}B^T \\ C^TQC & A^T \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = J \qquad (3.22)$$

Equating the (2,1) elements on each side of (3.22),

$$(-T_{21}\ A + T_{22}\ C^TQC)\ S_{11} + (T_{21}\ BR^{-1}B^T + T_{22}\ A^T)\ S_{21} = 0$$

or

$$-T_{21}\ AS_{11} + T_{22}\ A^T\ S_{21} + T_{21}\ BR^{-1}B^T\ S_{21} + T_{22}\ C^TQC\ S_{11} = 0 \qquad (3.23)$$

Substituting $T_{21} = -\ T_{22}\ S_{21}\ S_{11}^{-1}$ into (3.23),

$$T_{22}S_{21}S_{11}^{-1} + T_{22}A^TS_{21} - T_{22}S_{21}S_{11}^{-1}BR^{-1}B^TS_{21} + T_{22}C^TQC\ S_{11} = 0 \qquad (3.24)$$

Premultiplying by T_{22}^{-1} and postmultiplying by S_{11}^{-1},

$$S_{21}S_{11}^{-1}A + A^TS_{21}S_{11}^{-1} - S_{21}S_{11}^{-1}BR^{-1}B^TS_{21}S_{11}^{-1} + C^TQC = 0 \qquad (3.25)$$

which is to be proved.

4. COMPUTATIONAL METHODS AND NUMERICAL RESULTS

The major three steps for computing the solution of the matrix Riccati equation are as follows:

1) Compute eigenvalues and eigenvectors of W,

2) Compute principal vectors of W if W is defective,

3) Compute $S_{21} S_{11}^{-1}$ from S.

For computing the eigensystem, we used two methods: (i) the QR algorithm [9] with an inverse iteration routine [10], and (ii) Jacobi-like method [11] for finding the eigenvalues and eigenvectors. Their performances are satisfactory and two numerical test examples are given in Test Result 1 and Test Result 3. When the matrix W is nondiagonable, Test Result 2, the first method requires the evaluation of the principal vectors as well [12]. This poses some numerical difficulties as we are essentially trying to solve the ill-conditioned system of equation

$$(W - \tilde{\lambda}I)^k x_{j+1} = x_j, \; k > 1$$

where $\tilde{\lambda}$ is an approximation of true eigenvalues λ. Thus, the Jacobi-like method is more attractive in this case for providing an approximate eigensystem that is suitable for engineering purposes.

Appendix I contains the listings of the ALGOL programs used on the B-6500. Appendix II contains the listings of two ILLIAC IV programs, namely, the inverse iteration routine and a routine for solving a system of complex linear equations, both written in GLYPNIR. ILLIAC IV programs for the QR algorithm [13, 14] and the Jacobi-like algorithms [15, 16] are already available.

Test Result 1.

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The numerical example was tested by the QR algorithm and the inverse iteration method. The ten distinct eigenvalues of W are

$\pm\ 1,$

$\pm\ 1.728760477185 \pm 1.577533767573i,$

$\pm\ 1.353195784046 \pm 1.153749899284i.$

We choose the five eigenvectors corresponding to the five eigenvalues with positive real parts and computing $K = S_{21}\ S_{11}^{-1}$ from them gives the solution K as

$$\begin{matrix} 1.262782609 & 2.494009759 & -0.819173651 & 0.668267901 & -0.443608958 \\ 2.494009759 & 7.435451164 & -1.825741858 & 1.122910432 & -0.668267901 \\ -0.819173651 & -1.825741858 & 1.638347303 & 1.825741858 & -0.819173651 \\ 0.668267901 & 1.122910432 & 1.825741858 & 7.435451164 & -2.494009759 \\ -0.443608958 & -0.668267901 & -0.819173651 & -2.494009759 & 1.262782609 \end{matrix}$$

All the eigenvalues of $G = A - BR^{-1}B^TK$ have negative real parts and hence the above solution K is acceptable.

Test Result 2.

$$A = \begin{bmatrix} -3 & 2 \\ -2 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$P=Q= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad R = \begin{bmatrix} 1 \end{bmatrix}$$

The matrix

$$W = \begin{bmatrix} 3 & -2 & 0 & 0 \\ 2 & -1 & 0 & 1 \\ 0 & 0 & -3 & -2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

is defective, that is, W has four eigenvalues ±1 (double) and each +1 and -1 correspond to a quadratic elementary divisor, respectively. This example was tested both by Jacobi-like method and by a combination of the QR algorithm and the inverse iteration method.

The Jacobi-like method gives K as

$$\begin{matrix} -10.0000084 & 6.0000043 \\ 6.0000034 & -4.0000027 \end{matrix}$$

and the QR and inverse iteration method give K as

$$\begin{matrix} -9.9999917 & 5.9999947 \\ 5.9999953 & -3.9999983 \end{matrix}$$

The exact solution is

$$\begin{bmatrix} -10 & 6 \\ 6 & -4 \end{bmatrix}$$

For computing the principal vector by the QR and inverse iteration, we chose the perturbation for the true eigenvalues as much as 10^{-6}.

Test Result 3.

$$A = \begin{bmatrix} 6 & 4 & 4 & 1 \\ 4 & 6 & 1 & 4 \\ 4 & 1 & 6 & 4 \\ 1 & 4 & 4 & 6 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C=R= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad P=Q= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The exact eigenvalues of W are

$$\pm 15, \pm 5, \pm 5, \pm 1$$

Since the matrix W is derogatory, +5 and -5 correspond to a two-dimensional subspace, respectively. The computed eigenvalues by Jacobi-like method are

14.99999999930	-14.9999999984
4.99999999949	4.99999999981
-4.99999999982	-4.99999999948
0.99999999997	-0.99999999995

The computed solution K is given by

-1.999999999811	1.999999999942	1.999999999913	-2.000000000015
1.999999999884	-2.000000000015	-2.000000000000	2.000000000087
2.000000000276	-2.000000000407	-2.000000000378	2.000000000480
-2.000000000364	2.000000000509	2.000000000480	-2.000000000568

in which the absolute error is less than 10^{-9}.

LIST OF REFERENCES

[1] M. Athans and P. L. Falb, Optimal Control, McGraw Hill, New York, 1966

[2] R. E. Kalman, Contributions to the Theory of Optimal Control, Boll. Soci. Mat. Mexicana, 5 (1960), pp. 102-119

[3] R. S. Bucy and P. D. Joseph, Filtering for Stochastic Processes with Applications to Guidance, Interscience, New York, 1968

[4] A. G. J. MacFarlane, An Eigenvector Solution of the Optimal Linear Regulator Problem, J. Electronics and Control, 14 (1963), pp. 643-645

[5] J. J. O'donnell, Asymptotic Solution of the Matrix Riccati Equation of Optimal Control, Proc. Fourth Annual Allerton Conf., 1966, pp. 577-586

[6] J. E. Potter, Matrix Quadratic Solutions, J. SIAM Appl. Math., 14 (1966), pp. 496-501

[7] K. Martensson, On the Matrix Riccati Equation, Inf. Sci., 3 (1971), pp. 17-49

[8] D. L. Slotnick, et al, The ILLIAC IV Computer, IEEE Trans. on Computer. C-17 (1968), pp. 746-757

[9] R. S. Martin, G. Peters, and J. H. Wilkinson, The QR Algorithm for Real Hessenberg Matrices, Num. Math., 14 (1970), pp. 219-231

[10] J. M. Varah, The Calculation of the Eigenvectors of a General Complex Matrix by Inverse Iteration, Math. Comp., 22 (1968), pp. 785-791

[11] P. J. Eberlein and J. Boothroyd, Solution to the Eigenproblem by a Norm Reducing Jacobi Type Method, Num. Math., 11 (1968), pp. 1-12

[12] H. J. Bowdler, R. S. Martin, G. Peters, and J. H. Wilkinson, Solution of Real and Complex Systems of Linear Equations, Num. Math., 8 (1966), pp. 217-234

[13] M. Ogura, A Parallel Scheme for Reducing a Real Matrix to the Upper Hessenberg Form, Center for Advanced Computation Document No. 11, University of Illinois, 1971

[14] M. Ogura, The QR-Algorithm, Center for Advanced Computation Document No. 12, Univ. of Ill., 1971

[15] W. Bernhard, ILLIAC IV Codes for Jacobi and Jacobi-like Algorithms, Center for Advanced Computation Document No. 19, Univ. of Ill., 1971

[16] A. H. Sameh, On Jacobi and Jacobi-Like Algorithms for a Parallel Computer, Math. Comp., vol. 25 (1971), pp. 579-590

APPENDIX I

ALGOL PROGRAMS

```
% TWO ALGOL PROCEDURES, RICCATI1 AND RICCATI2, FOR COMPUTING THE
% SOLUTION OF THE MATRIX RICCATI EQUATION ARE LISTED BELOW. FOR
% "RICCATI1", THE PROCEDURE "EIGEN" (BY EBERLEIN AND BOOTHROYD) WAS
% USED WHILE A COMBINATION OF "HQR" (BY MARTIN, ET AL) AND
% "INVITERATION" WAS APPLIED TO "RICCATI2". LINEAR EQUATION SOLVER
% (BY BOWDLER, ET AL) WAS ALSO USED TO PROVIDE THE INVERSE OF A MATRIX
% AND TO CALCULATE THE PRINCIPAL VECTORS FROM ((W-LAMBDA*I)**K)*X=0.


PROCEDURE RICCATI1(N,L,M,A,B,C,P,Q,R,TMX,W,K);
VALUE N,L,M,TMX;
INTEGER N,L,M,TMX;
ARRAY A,B,C,P,Q,R,W,K[1,1];
COMMENT SOLVES THE MATRIX RICCATI EQUATION
           KA+(AT)K-KB(RINV)(BT)K+(CT)QC=0.
        A IS N*N, B IS N*L, C IS M*N, P AND Q ARE M*M, R IS L*L
        MATRICES, RESPECTIVELY. FORM 2N*2N MATRIX W AND COMPUTE
        EIGENVALUES AND EIGENVECTORS OF W, WHERE FOUR N*N BLOCKS
        OF W ARE (1,1)=-A, (1,2)=B(RINV)(BT), (2,1)=(CT)QC, AND
        (2,2)=(AT). REARRANGE EIGENVECTOR(PRINCIPAL VECTOR) MATRIX
        SO THAT THE FIRST N COLUMNS CORRESPOND TO TO THE EIGENVALUES
        WITH POSITIVE REAL PARTS. DENOTING THE UPPER HALF OF N COLUMN
        VECTORS BY SMTX1 AND THE LOWER HALF BY SMTX2, THE SOLUTION
        K IS THEN GIVEN BY K=(SMTX2)(SMTX1INV). THE PROCEDURE
        "EIGEN" IS USED FOR EIGENPROBLEM OF W;

BEGIN COMMENT FIRST DECLARE PROCEDURES;

PROCEDURE SETUP(N,L,M,A,B,C,Q,RINV,W);
VALUE N,L,M;  INTEGER N,L,M;
REAL ARRAY A,B,C,Q,RINV,W[1,1];
COMMENT  THE PROCEDURE SETS UP THE 2N*2N MATRIX W. RINV IS INVERSE OF R;
BEGIN
  REAL ARRAY RINVBT[1:L,1:N],QC[1:M,1:N];
  REAL SUM;
  INTEGER I,J,K;
  FOR I:=1 STEP 1 UNTIL N DO
  FOR J:=1 STEP 1 UNTIL N DO
    BEGIN
      W[I,J]:=-A[I,J];
      W[N+I,N+J]:=A[J,I];
    END;
  FOR I:=1 STEP 1 UNTIL L DO
  FOR J:=1 STEP 1 UNTIL N DO
    BEGIN SUM:=0.0;
      FOR K:=1 STEP 1 UNTIL L DO
        SUM:=SUM+RINV[I,K]*B[J,K];
      RINVBT[I,J]:=SUM;
    END;
  FOR I:=1 STEP 1 UNTIL N DO
  FOR J:=1 STEP 1 UNTIL N DO
    BEGIN SUM:=0.0;
      FOR K:=1 STEP 1 UNTIL L DO
        SUM:=SUM+B[I,K]*RINVBT[K,J];
      W[I,N+J]:=SUM;
    END;
  FOR I:=1 STEP 1 UNTIL M DO
  FOR J:=1 STEP 1 UNTIL N DO
    BEGIN SUM:=0.0;
      FOR K:=1 STEP 1 UNTIL M DO
        SUM:=SUM+Q[I,K]*C[K,J];
      QC[I,J]:=SUM;
    END;
  FOR I:=1 STEP 1 UNTIL N DO
  FOR J:=1 STEP 1 UNTIL N DO
    BEGIN SUM:=0.0;
      FOR K:=1 STEP 1 UNTIL M DO
        SUM:=SUM+C[K,I]*QC[K,J];
      W[N+I,J]:=SUM;
    END;
END OF SETUP;

COMMENT INSERT THE PROCEDURES INNPROD,DECOMPOSE, AND ACCSOLVE;

PROCEUURE INVERSE(N,A,X);
VALUE N; INTEGER N;
DOUBLE ARRAY A,X[1,1];
COMMENT THE PROCEDURE COMPUTES THE INVERSE OF AN N*N REAL
        UNSYMMETRIC MATRIX A, USING THE PROCEDURES "DECOMPUSE" AND
        "ACCSOLVE", WRITING ON X;
BEGIN
  INTEGER I,J,IT,D2;
  REAL D1;
  ARRAY AA,BB,B[1:N,1:N],P[1:N];
    FUR I:=1 STEP 1 UNTIL N DO
    FUR J:=1 STEP 1 UNTIL N DO
      IF J=I THEN B[I,J]:=1.0 ELSE B[I,J]:=0.0;
    FUR I:=1 STEP 1 UNTIL N DO
    FUR J:=1 STEP 1 UNTIL N DO
      AA[I,J]:=A[I,J];
    DECOMPOSE(N,AA,D1,D2,P);
    ACCSULVE(N,N,A,AA,P,B,X,BB,IT);
END INVERSE;

COMMENT INSERT THE PROCEDURE EIGEN;

PROCEUURE MULT(N1,N2,N3,A,B,C);
VALUE N1,N2,N3; INTEGER N1,N2,N3;
DOUBLE ARRAY A,B,C[1,1];
BEGIN
  INTEGER I,J,K;
  REAL SUM;
  FOR I:=1 STEP 1 UNTIL N1 DO
  FOR J:=1 STEP 1 UNTIL N3 DO
  BEGIN SUM:=0.0;
    FUR K:=1 STEP 1 UNTIL N2 DO
      SUM:=SUM+A[I,K]*B[K,J];
    C[I,J]:=SUM;
  END;
END MULT;

  ARRAY RINV,SMTX1,SMTX2[1:N,1:N],WW,TEMP,VECT[1:N+N,1:N+N];
  INTEGER ARRAY SEARCH[1:N];
  INTEGER N2,I,J,INDEX;
  N2:=N+N;
  INVERSE(L,R,RINV);
  SETUP(N,L,M,A,B,C,Q,RINV,W);
  FOR I:=1 STEP 1 UNTIL N2 DO
  FOR J:=1 STEP 1 UNTIL N2 DO
    TEMP[I,J]:=W[I,J];
  EIGEN(N2,TMX,TEMP,VECT);
  COMMENT SEARCH N COLUMNS OF VECT CORRESPONDING TO N EIGENVALUES
          WITH POSITIVE REAL PARTS, SMTX1 AND SMTX2 ARE ITS UPPER
          AND LOWER HALVES, RESPECTIVELY;
  INDEX:=0;
  FOR I:=1 STEP 1 UNTIL N2 DO
    IF TEMP[I,I] GTR 0 THEN
    BEGIN
      INDEX:=INDEX+1;
      SEARCH[INDEX]:=I;
    END;
  FOR J:=1 STEP 1 UNTIL N DO
  FOR I:=1 STEP 1 UNTIL N DO
  BEGIN
    SMTX1[I,J]:=VECT[I,SEARCH[J]];
    SMTX2[I,J]:=VECT[N+I,SEARCH[J]];
  END;
  INVERSE(N,SMTX1,TEMP);
  MULI(N,N,N,SMTX2,TEMP,K);
END RICCATI1;


PROCEUURE RICCATI2(N,L,M,A,B,C,P,Q,R,W,K,DEFECT,DEROGAIE);
VALUE N,L,M;
INTEGER N,L,M;
ARRAY A,3,C,P,Q,R,W,K[1,1];
INTEGER ARRAY DEFECT,DERUGATE[1];
COMMENT SOLVES THE MATRIX RICCATI EQUATION
           KA+(AT)K-KB(RINV)(BT)K+(CT)QC=0.
        A IS N*N, B IS N*L, C IS M*N, P AND Q ARE M*M, R IS L*L
        MATRICES, RESPECTIVELY. FORM 2N*2N MATRIX W AND COMPUTE
        EIGENVALUES ANU EIGENVECTORS OF W, WHERE FOUR N*N BLOCKS
        OF W ARE (1,1)=-A, (1,2)=B(RINV)(BT), (2,1)=(CI)QC, AND
        (2,2)=(AT). REARRANGE EIGENVECTOR(PRINCIPAL VECTOR) MATRIX
        SO THAT THE FIRST N COLUMNS CORRESPOND TO TO THE EIGENVALUES
        WITH POSITIVE REAL PARTS. DENOTING THE UPPER HALF UF N COLUMN
        VECTORS BY SMTX1 AND THE LOWER HALF BY SMTX2, THE SOLUTION
        K IS THEN GIVEN BY K=(SMTX2)(SMTX1INV). THE PRUCEDURES
        "HSHLD", "HQR", AND "INVITERATION" ARE USED FOR EIGENPROBLEM
        OF W;

BEGIN COMMENT FIRST DECLARE PROCEDURES;

PROCEUURE SETUP(N,L,M,A,B,C,Q,RINV,W);
VALUE N,L,M;  INTEGER N,L,M;
REAL ARRAY A,B,C,Q,RINV,W[1,1];
COMMENT THE PROCEDURE SETS UP THE 2N*2N MATRIX W, RINV IS INVERSE OF R;
BEGIN
  REAL ARRAY RINVBT[1:L,1:N],QC[1:M,1:N];
  REAL SUM;
  INTEGER I,J,K;
  FOR I!=1 STEP 1 UNTIL N DO                                          00017200?
  FOR J;=1 STEP 1 UNTIL N DO                                          00017300?
    BEGIN                                                             00017400?
      W[I,J]!=-A[I,J]!                                                00017500?
      W[N+I,N+J]!=A[J,I]!                                             00017600?
    END!                                                              00017700?
  FOR I!=1 STEP 1 UNTIL L DO                                          00017800?
  FOR J!=1 STEP 1 UNTIL N DO                                          00017900?
    BEGIN SUM!=0.0!                                                   00018000?
      FOR K;=1 STEP 1 UNTIL L DO                                      00018100?
        SUM;=SUM+RINV[I,K]*B[J,K]!                                    00018200?
      RINVBT[I,J]!=SUM!                                               00018300?
    END;                                                              00018400?
  FOR I;=1 STEP 1 UNTIL N DO                                          00018500?
  FOR J!=1 STEP 1 UNTIL N DO                                          00018600?
    BEGIN SUM!=0.0!                                                   00018700?
      FOR K;=1 STEP 1 UNTIL L DO                                      00018800?
        SUM;=SUM+B[I,K]*RINVBT[K,J]!                                  00018900?
      W[I,N+J]!=SUM!                                                  00019000?
    END;                                                              00019100?
  FOR I!=1 STEP 1 UNTIL M DO                                          00019200?
  FOR J;=1 STEP 1 UNTIL N DO                                          00019300?
    BEGIN SUM!=0.0!                                                   00019400?
      FOR K;=1 STEP 1 UNTIL M DO                                      00019500?
        SUM;=SUM+Q[I,K]*C[K,J]!                                       00019600?
      QC[I,J]!=SUM!                                                   00019700?
    END!                                                              00019800?
  FOR I!=1 STEP 1 UNTIL N DO                                          00019900?
  FOR J;=1 STEP 1 UNTIL N DO                                          00020000?
    BEGIN SUM!=0.0!                                                   00020100?
      FOR K;=1 STEP 1 UNTIL M DO                                      00020200?
        SUM;=SUM+C[K,I]*QC[K,J]!                                      00020300?
      W[N+I,J]!=SUM!                                                  00020400?
    END;                                                              00020500?
END OF SETUP!                                                         00020600?
                                                                      00020700?
COMMENT INSERT THE PROCEDURES INNPROD,DECOMPOSE, AND ACCSOLVE;        00020800?
                                                                      00020900?
PROCEDURE INVERSE(N,A,X)!                                             00021000?
VALUE N! INTEGER N!                                                   00021100?
DOUBLE ARRAY A,X[1,1]!                                                00021200?
COMMENT THE PROCEDURE COMPUTES THE INVERSE OF AN N*N REAL             00021300?
        UNSYMMETRIC MATRIX A, USING THE PROCEDURES "DECOMPOSE" AND    00021400?
        "ACCSOLVE", WRITING ON X!                                     00021500?
BEGIN                                                                 00021600?
  INTEGER I,J,IT,D2!                                                  00021700?
  REAL D1;                                                            00021800?
  ARRAY AA,BB,B[1:N,1:N],P[1:N];                                      00021900?
    FOR I!=1 STEP 1 UNTIL N DO                                        00022000?
    FOR J!=1 STEP 1 UNTIL N DO                                        00022100?
      IF J=I THEN B[I,J]!=1.0 ELSE B[I,J]!=0.0!                       00022200?
    FOR I;=1 STEP 1 UNTIL N DO                                        00022300?
    FOR J!=1 STEP 1 UNTIL N DO                                        00022400?
      AA[I,J]!=A[I,J]!                                                00022500?
    DECOMPOSE(N,AA,D1,D2,P)!                                          00022600?
    ACCSOLVE(N,N,A,AA,P,B,X,BB,IT);                                   00022700?
END INVERSE!                                                          00022800?
```

```
COMMENT INSERT THE PROCEDURES HSHLD AND HQR;

PROCEDURE MULT(N1,N2,N3,A,B,C);
VALUE N1,N2,N3; INTEGER N1,N2,N3;
DOUBLE ARRAY A,B,C[1,1];
BEGIN
  INTEGER I,J,K;
  REAL SUM;
  FOR I:=1 STEP 1 UNTIL N1 DO
  FOR J:=1 STEP 1 UNTIL N3 DO
  BEGIN SUM:=0.0;
    FUR K:=1 STEP 1 UNTIL N2 DO
      SUM:=SUM+A[I,K]*B[K,J];
    C[I,J]:=SUM;
  END;
END MULT;

PROCEUURE INVITERATION(N,W,X,KK,P);
VALUE N; INTEGER N,KK;
DOUBLE ARRAY W,X[1,1];
ARRAY P[1];
COMMENT SOLVES  WX=0 BY INVERSE ITERATION, WHERE W IS AN N*N REAL
        MATRIX, WRITING SOLUTION ON X. THUS, THE PROCEDURE COMPUTES
        THE EIGENVECTORS CORRESPONDING TO THE GIVEN EIGENVALUES.
        N IS THE ORDER OF W AND KK IS THE NUMBER OF ITERATIONS
        NEEDED FOR ITERATING EACH EIGENVECTOR;
BEGIN
  INTEGER I,J,D2,R,L,MAXINDEX;
  DOUBLE D1,MAXVAL;
  DOUBLE ARRAY WW[1:N,1:N],B,BB[1:N,1:1];
  LABEL INVIT,OUT;
  FOR I:=1 STEP 1 UNTIL N DO
  FOR J:=1 STEP 1 UNTIL N DO
    WW[I,J]:=W[I,J];
  DECUMPOSE(N,WW,D1,D2,P);
  FOR I:=1 STEP 1 UNTIL N DO
    B[I,1]:=1.0;
  KK:=1;  R:=1;
INVIT:
  ACCSOLVE(N,R,W,WW,P,B,X,BB,L);
  MAXVAL:=0;
  FOR I:=1 STEP 1 UNTIL N DO
    IF ABS(X[I,1]) GTR MAXVAL THEN
      BEGIN
        MAXVAL:=ABS(X[I,1]);
        MAXINDEX:=I;
      END;
    MAXVAL:=X[MAXINDEX,1];
  FOR I:=1 STEP 1 UNTIL N DO
    X[I,1]:=X[I,1]/MAXVAL;
  FOR I:=1 STEP 1 UNTIL N DO
    IF ABS(B[I,1]-X[I,1]) GTR @-12 THEN
    BEGIN
      IF KK GTR 10 THEN GO TO OUT;
      FOR J:=1 STEP 1 UNTIL N DU
        B[J,1]:=X[J,1];
      KK:=KK+1;
      GO TO INVIT;
    END;
OUT:
END INVRESEIT;

   INTEGER I,J,JJ,CNT,N2,INDEX,ITCOUNT;
   ARRAY TEMP,ORGW,SMTX,VECTOR[1:N+N,1:N+N],RINV,SMTX1,SMTX2[1:N,1:N],
         EIGR,EIGI,INTCH[1:N+N];
   REAL PERTURB,MPLIER;
   INTEGER ARRAY SEARCH[1:N];
   N2:=N+N;
   INVERSE(L,R,RINV);
   SETUP(N,L,M,A,B,C,Q,RINV,W);
   FOR I:=1 STEP 1 UNTIL N2 DO
   FOR J:=1 STEP 1 UNTIL N2 DO
     TEMP[I,J]:=W[I,J];
   HSHLD(1,N2,TEMP);
   HQR(N2,TEMP,EIGR,EIGI,CNT);
   FOR I:=1 STEP 1 UNTIL N2 DO
   FOR J:=1 STEP 1 UNTIL N2 DO
     ORGW[I,J]:=W[I,J];
   COMMENT COMPUTE EIGENVECTORS OR PRINCIPAL VECTORS;
   FOR JJ:=1 STEP 1 UNTIL N2 DO
   BEGIN
     FOR I:=1 STEP 1 UNTIL N2 DO
     FOR J:=1 STEP 1 UNTIL N2 DO
       W[I,J]:=ORGW[I,J];
     FOR I:=1 STEP 1 UNTIL N2 DO
       W[I,I]:=W[I,I]-EIGR[JJ]+PERTURB;
     IF JJ=DEFECT[JJ] THEN
       BEGIN
         FOR I:=1 STEP 1 UNTIL N2 DO
           W[I,I]:=W[I,I]+PERTURB*MPLIER;
         MULT(N2,N2,N2,W,W,TEMP);
         FOR I:=1 STEP 1 UNTIL N2 DO
         FOR J:=1 STEP 1 UNTIL N2 DO
           W[I,J]:=TEMP[I,J];
       END;
     IF JJ=DEROGATE[JJ] THEN
       FOR I:=1 STEP 1 UNTIL N2 DO
         W[I,I]:=W[I,I]+PERTURB*JJ;
     INVITERATION(N2,W,VECTOR,ITCOUNT,INTCH);
     FOR I:=1 STEP 1 UNTIL N2 DO
     FOR J:=1,2 DO
       SMTX[I,JJ]:=VECTOR[I,1];
   END JJ;
   COMMENT SEARCH N COLUMNS OF SMTX CORRESPONDING TO N EIGENVALUES
           WITH POSITIVE REAL PARTS, SMTX1 AND SMTX2 ARE ITS UPPER
           AND LOWER HALVES, RESPECTIVELY;
   INDEX:=0;
   FOR I:=1 STEP 1 UNTIL N2 DO
     IF EIGR[I] GTR 0 THEN
     BEGIN
       INDEX:=INDEX+1;
       SEARCH[INDEX]:=I;
     END;
```

```
  FOR J:=1 STEP 1 UNTIL N DO
  FOR I:=1 STEP 1 UNTIL N DO
  BEGIN
    SMTX1[I,J]:=SMTX[I,SEARCH[J]];
    SMTX2[I,J]:=SMTX[N+I,SEARCH[J]];
  END;
  INVERSE(N,SMTX1,TEMP);
  MULT(N,N,N,SMTX2,TEMP,K);
END RICCATI2;
```

APPENDIX II

ILLIAC IV PROGRAMS

```
% TWO ILLIAC IV GLYPNIR PROGRAMS, "CLINSYS" AND "INVITERATION" ARE
% LISTED BELOW, "CLINSYS" IS USED TO OBTAIN THE INVERSE OF A MATRIX
% AND THE SOLUTION OF A COMPLEX SYSTEM OF UNSYMMETRIC LINEAR EQUATIONS.
% "INVITERATION" IS USED TO CALCULATE THE PRINCIPAL VECTORS OF DEGREE
% K FROM ((W-LAMBDA*I)**K)*X=0.


SUBROUTINE CLINSYS(CINT N,CINT R,PCPOINT A,PCPOINT X,PCPOINT B);
BEGIN % THE SUBROUTINE SOLVES A COMPLEX SYSTEM OF UNSYMMETRIC LINEAR
      % EQUATIONS AX=B. A IS N*2N AND B IS N*2R MATRICES, RESPECTIVELY.
      % THEIR GENERAL ELEMENTS ARE A[I,2J-1]+I*A[I,2J] AND B[I,2J-1]+
      % I*B[I,2J], RESPECTIVELY, R IS THE NUMBER OF RIGHT-HAND SIDES,

  SUBROUTINE INNPROD(CREAL C1,CREAL C2,PREAL AK,PREAL BK,CREAL D1,
                                                    CREAL D2);
  BEGIN % ACCUMULATES THE SUM OF PRODUCTS AND ADDS IT 10 THE INITIAL
        % VALUE (C1,C2).
    CREAL S12;
    S12:=C1+C2;
    S12:=S12+ROWSUM(AK*BK);
    D1:=S12;
    D2:=S12-D1;
  END;

  SUBROUTINE CDECOMPOSE(CINT N,PCPOINT A,CREAL DETR,CREAL DETI,
                        CINT DETE,CNPOINT INT);
  BEGIN % THE COMPLEX UNSYMMETRIC MATRIX A IS DECOMPOSED INTO LU,
        % WHERE L IS LOWER TRIANGULAR AND U IS UNIT UPPER TRIANGULAR
        % MATRICES, AND OVERWRITTEN ON A. A IS STORED IN N*2N ARRAY
        % AND ITS GENERAL ELEMENTS ARE A[I,2J-1]+I*A[I,2J], INT[I]
        % KEEPS THE RECORD OF ANY INTERCHANGES MADE TO THE ROWS OF
        % A. THE DETERMINANT (DETR+I*DETI)*2**DETE IS ALSO COMPUTED.
    CINT I,J,K,L,P,PP;
    PREAL VECTOR AT[64];
    PREAL ZZ;
    CREAL X,Y,Z,V,W,H,HH;
    LABEL FAIL;
    MUDE:=TRUE;
    MUDE:=TRUE AND PEN LSS N+N;
    LUUP I:=1,1,N DO
      INNPROD(0,0,A[I],A[I],INT[I],W);
    DETR:=1; DETI:=0; DETE:=0;
    LUOP K:=1,1,N DO
    BEGIN
      L:=K; P:=K+K; PP:=P-1; Z:=0;
      LOOP I:=K,1,N DO
      BEGIN
        MODE:=TRUE;
        MODE:=TRUE AND PEN LSS N+N;
        INNPROD(-GRABONE(A[I],PP-1),0,A[I],AT[PP],H,HH);
        INNPROD(-H,-HH,RTL(1,,A[I]),AT[P],X,HH);
        INNPROD(-GRABONE(A[I],P-1),0,RTL(1,,A[I]),AT[PP],H,HH);
        INNPROD(H,HH,A[I],AT[P],H,HH);
        Y:=-H;
        MODE:=TRUE;
        MODE:=TRUE AND PEN EQL PP-1;  A[I]:=X;
        MODE:=TRUE;
      MODE:=TRUE AND PEN EQL P-1;  A[I]:=Y;
      MODE:=TRUE;
      MODE:=TRUE AND PEN=I;
      X:=(X*X+Y*Y)/INT[I];
      IF X GTR Z THEN
      BEGIN Z:=X; L:=I; END;
    END;
    MODE:=TRUE; MODE:=TRUE AND PEN LSS N+N;
    IF L NEQ K THEN
    BEGIN DETR:=-DETR;
          DETI:=-DETI;
          ZZ:=A[K];
          A[K]:=A[L];
          A[L]:=ZZ;
          INT[L]:=INT[K];
    END;
    INT[K]:=L;
    X:=GRABONE(A[K],PP-1);
    Y:=GRABONE(A[K],P-1);
    Z:=X*X+Y*Y;
    W:=X*DETR-Y*DETI;
    DETI:=X*DETI+Y*DETR;
    DETR:=W;
    IF ABS(DETR) GTR ABS(DETI)
      THEN W:=DETR ELSE W:=DETI;
    IF W =0 THEN BEGIN DETE:=0; GO TO FAIL; END;
    BEGIN LABEL L1;
L1: IF ABS(W) GEQ 1 THEN
    BEGIN W:=W*0.0625;
          DETI:=DETI*0.0625;
          DETE:=DETE+4;
          GO TO L1;
    END;
    END;
    BEGIN LABEL L2;
L2: IF ABS(W) LSS 0.0625 THEN
    BEGIN W:=W*16;
          DETR:=DETR*16;
          DETI:=DETI*16;
          DETE:=DETE-4;
          GO TO L2;
    END;
    END;
    LOOP J:=K+1,1,N DO
    BEGIN
      P:=J+J; PP:=P-1;
      MODE:=TRUE;
      MODE:=TRUE AND PEN LSS K+K-2;
      INNPROD(-GRABONE(A[K],PP-1),0,A[K],AT[PP],H,HH);
      INNPROD(-H,-HH,RTL(1,,A[K]),AT[P],V,HH);
      INNPROD(-GRABONE(A[K],P-1),0,RTL(1,,A[PP]),AT[PP],H,HH);
      INNPROD(H,HH,A[K],AT[P],H,HH);
      W:=-H;
      MODE:=TRUE;
      MODE:=TRUE AND PEN=PP-1;
        A[K]:=(V*X+W*Y)/Z;
      MODE:=TRUE;
```

```
        MODE:=TRUE AND PEN=P-1;
          A[K]:=(W*X-V*Y)/Z;
      END;
    END;
FAIL:
END; %OF CDECOMPOSE.

SUBROUTINE CACCSOLVE(CINT N,CINT R,PCPOINT A,PCPOINT AA,CNPOINT P,
                     PCPOINT B,PCPOINT X,PCPOINT BB,CINT L);
BEGIN % SOLVES AX=B, WHERE A IS AN N*2N COMPLEX UNSYMMETRIC AND
      % B IS AN N*2R COMPLEX MATRICES, RESPECTIVELY. AA IS LU
      % DECOMPOSITION PRODUCED BY "CDECOMPOSE". THE RESIDUALS
      % BB=B-AX ARE ALSO CALCULATED AND AD=BB IS SOLVED, OVER-
      % WRITING D ON BB. L IS THE NUMBER OF ITERATIONS.

  SUBROUTINE CSOLVE(CINT N,CINT R,PCPOINT A,CNPOINT INT,
                     PCPOINT B);
  BEGIN
    CINT I,J,K,KK,P,PP;
    PREAL VECTOR BT[64];
    CREAL X,Y,Z,Z1,Z2,H,HH;
    PREAL VECTOR C[64];
    BOOLEAN AMODE;
    LOOP I:=1,1,N DO
      IF INT[I] NEQ I THEN
        LOOP J:=R+R,-1,1 DO
        BEGIN
          X:=GRABONE(B[I],J-1);
          MODE:=TRUE AND PEN=J-1;
          B[I]:=GRABONE(B[INT[I]],J-1);
          B[INT[I]]:=X;
        END;
    LOOP K:=R+R,-2,2 DO
    BEGIN
      KK:=K-1;
      LOOP I:=1,1,N DO
      BEGIN
        MODE:=TRUE AND PEN LSS I+I-2;
        INNPROD(-GRABONE(B[I],KK-1),0,A[I],BT[KK],H,HH);
        INNPROD(-H,-HH,RTL(1,,A[I]),BT[K],X,HH);
        INNPROD(-GRABONE(B[I],K-1),0,RTL(1,,A[I]),BT[KK],H,HH);
        INNPROD(H,HH,A[I],BT[K],H,HH);
        Y:=-H;
        P:=I+I; PP:=P-1;
        Z1:=GRABONE(A[I],PP-1);
        Z2:=GRABONE(A[I],P-1);
        Z:=Z1*Z1+Z2*Z2;
        MODE:=TRUE AND PEN=KK;
          B[I]:=(X*Z1+Y*Z2)/Z;
        MODE:=TRUE AND PEN=K;
          B[I]:=(Y*Z1-X*Z2)/Z;
      END;
      LOOP I:=N,-1,1 DO
      BEGIN
        AMODE:= TRUE AND PEN GEQ N;
        MODE:=AMODE;
        INNPROD(-GRABONE(B[I],KK-1),0,A[I],BT[KK],H,HH);
```

```
          INNPROD(-H,-HH,RTL(1,,A[I]),BT[K],H,HH);
          MODE:=TRUE AND PEN=KK-1; C[I]:=H;
          MODE:=AMODE;
          INNPROD(-GRABONE(B[I],K-1),0,RTL(1,,A[I]),BT[KK],H,HH);
          INNPROD(H,HH,A[I],BT[K],H,HH);
          MODE:=TRUE AND PEN=K-1; C[I]:=-H;
          MODE:=TRUE AND (PEN=KK-1 OR PEN=K-1); B[I]:=C[I];
        END;
      END;
    END; % OF CSOLVE.

    CINT I,J,K,D2,C,CC;
    CREAL E,D0,D1,XMAX,BBMAX,EPS,E1,E2,H,HH;
    PREAL VECTOR XT[64];
    BOOLEAN AMODE;
    LABEL L3,ILL;
    EPS:=1.0@-10;
    AMODE:=PEN GEQ N;
    MODE:=AMODE;
    LOOP I:=1,1,N DO
    BEGIN X[I]:=0.0;
          BB[I]:=B[I];
    END;
    L:=0; D0:=0;
    MODE:=TRUE;
L3: CSOLVE(N,R,AA,P,BB);
    L:=L+1; D2:=0; D1:=0;
    MODE:=AMODE;
    LOOP I:=1,1,N DO
      X[I]:=X[I]+BB[I];
    LOOP J:=1,1,R DO
    BEGIN
      XMAX:=0;  BBMAX:=0;
      C:=J+J; CC:=C-1;
      LOOP I:=1,1,N DO
      BEGIN
        E1:=GRABONE(X[I],CC-1); E2:=GRABONE(X[I],C-1);
        E:=E1*E1+E2*E2;
        IF E GTR XMAX THEN XMAX:=E;
        E1:=GRABONE(BB[I],CC); E2:=GRABONE(BB[I],C);
        E:=E1*E1+E2*E2;
        IF E GTR BBMAX THEN BBMAX:=E;
        MODE:=TRUE;
        INNPROD(-GRABONE(B[I],CC),0,A[I],XT[CC],H,HH);
        INNPROD(-H,-HH,RTL(1,,A[I]),XT[C],H,HH);
        MODE:=TRUE AND PEN=CC-1; BB[I]:=H;
        MODE:=TRUE;
        INNPROD(-GRABONE(B[I],C),0,RTL(1,,A[I]),XT[C],H,HH);
        INNPROD(H,HH,A[I],XT[C],H,HH);
        MODE:=TRUE AND PEN=C-1; BB[I]:=-H;
      END;
      IF BBMAX/XMAX GTR D1 THEN D1:=BBMAX/XMAX;
      IF BBMAX GTR (2*EPS)*(2*EPS)*XMAX THEN D2:=1;
    END;
    IF D1 GTR D0*0.25 AND L NEQ 1 THEN GO TO ILL;
    D0:=D1;
    IF D2=1 THEN GO TO L3;
```

```
ILL:
  END; % OF CACCSOLVE.

  PREAL VECTOR AA,BB[64];
  CREAL VECTOR INTCH[64];
  CINT I,N,R,DETE,IT;
  CREAL DETR,DETI;
  LOOP I:=1,1,N DO
    AA[I]:=A[I];
  CDECOMPOSE(N,AA,DETR,DETI,DETE,INTCH);
  CACCSOLVE(N,R,A,AA,INTCH,B,X,BB,IT);
END; % OF CLINSYS.


SUBROUTINE INVITERATION(CINT N,PCPOINT W,PCPOINT X,CINT KK);
BEGIN % THE SUBROUTINE COMPUTES THE EIGENVECTORS AND PRINCIPAL
      % VECTORS BY SOLVING ((W-LAMBDA*I)**K)*X=0. THE SUBRUUTINES
      % "CDECOMPOSE" AND "CACCSOLVE" IN "CLINSYS" ARE USED.
  PE REAL SUBROUTINE SQRT AS RGA(PE REAL ARG AS RGA);
  BEGIN
$* CALL SQRT64();
  END; % OF SQRT.
    CINT I,J,II,DETE,R,L;
    CREAL DETR,DETI,MAXVAL,XX1,XX2;
    PREAL VECTOR W2,B,B2,BB,XX[64];
    CREAL VECTOR P,ROOT[64];
    PREAL TR1,TR2;
    LABEL INVIT,NEXT,FINIS;
      MODE:=PEN LSS N*N;
      LOOP I:=1,1,N DO W2[I]:=W[I];
      CDECOMPOSE(N,W2,DETR,DETI,DETE,P);
      MODE:=PEN LSS 2;
      LOOP I:=1,1,N DO
      BEGIN B2[I]:=1,0; B[I]:=B2[I]; END;
      KK:=1;
      R:=1;
INVIT:
      MODE:=PEN LSS N*N;
      CACCSOLVE(N,R,W,W2,P,B,X,BB,L);
      MODE:=PEN LSS 2;
      LOOP I:=1,1,N DO
      BEGIN
        ROOT[I]:=ROWSUM(X[I]*X[I]);
        ROOT[I]:=SQRT(ROOT[I]);
      END;
      MAXVAL:=ROOT[1];
      LOOP I:=2,1,N DO
        IF MAXVAL GEQ ROOT[I] THEN MAXVAL:=MAXVAL ELSE MAXVAL:=ROOT[I];
      LOOP I:=1,1,N DO
        IF ROOT[I]=MAXVAL THEN BEGIN II:=I; GO TO NEXT; END;
NEXT:
      XX1:=GRABONE(X[II],0);
      XX2:=-GRABONE(X[II],1);
      IF -XX2 NEQ 0 THEN
      BEGIN
        LOOP I:=1,1,N DO
        BEGIN
```

```
          MODE:=PEN LSS 2;
            TR1:=GRABONE(X[I],0)*XX1-GRABONE(X[I],1)*XX2;
            TR2:=GRABONE(X[I],0)*XX2+GRABONE(X[I],1)*XX1;
          MODE:=PEN=0;  XX[I]:=TR1;
          MODE:=PEN=1;  XX[I]:=TR2;
        END;
        MODE:=PEN LSS 2;
        XX1:=GRABONE(XX[I],0);
        LOOP I:=1,1,N DO X[I]:=XX[I]/XX1;
      END
      ELSE LOOP I:=1,1,N DO X[I]:=X[I]/XX1;
      LOOP I:=1,1,N DO
      BEGIN
        IF ABS(B2[I]-X[I]) GTR @-10 THEN
        BEGIN
          IF KK GTR 10 THEN GO TO FINIS;
          LOOP J:=1,1,N DO BEGIN B[I]:=X[I];
                                  B2[I]:=X[I];
                            END;
          KK:=KK+1;
          GO TO INVIT;
        END
        ELSE GO TO FINIS;
      END;
FINIS:
END; OF INVITERATION.
```

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

A NUMERICAL SOLUTION OF THE MATRIX RICCATI EQUATIONS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Research Report

5. AUTHOR(S) *(First name, middle initial, last name)*

Killion Noh

6. REPORT DATE

January 20, 1972

7a. TOTAL NO. OF PAGES

38

7b. NO. OF REFS

16

8a. CONTRACT OR GRANT NO.

DAHC04 72-C-0001

b. PROJECT NO.

ARPA Order 1899

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

CAC Document No. 24

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

UIUCDCS-R-72-498

10. DISTRIBUTION STATEMENT

Copies may be obtained from the address given in (1) above.

Distribution unlimited; approved for public release.

11. SUPPLEMENTARY NOTES

None

12. SPONSORING MILITARY ACTIVITY

U.S. Army Research Office-Durham
Durham, North Carolina

13. ABSTRACT

The eigenvector solution of the time-invariant matrix Riccati equation is discussed. The coefficient matrix of the canonical equation is allowed to have multiple eigenvalues, namely, the matrix could be either derogatory or defective. The solution of matrix Riccati equation is then calculated from a part of similarity transformation which should reduce the coefficient matrix to the Jordan canonical form.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Ordinary and Partial Differential Equations						
Matrix Algebra						
Nonlinear and Functional Equations						

BIBLIOGRAPHIC DATA SHEET

1. Report No. UIUCDCS-R-72-498

2.

3. Recipient's Accession No.

4. Title and Subtitle

A NUMERICAL SOLUTION OF THE MATRIX RICCATI EQUATIONS

5. Report Date

January 20, 1972

6.

7. Author(s)

Killion Noh

8. Performing Organization Rept. No.

9. Performing Organization Name and Address

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.

DAHC04 72-C-0001

12. Sponsoring Organization Name and Address

U.S. Army Research Office-Durham
Duke Station
Durham, North Carolina

13. Type of Report & Period Covered

Research

14.

15. Supplementary Notes

None

16. Abstracts

The eigenvector solution of the time-invariant matrix Riccati equation is discussed. The coefficient matrix of the canonical equation is allowed to have multiple eigenvalues, namely, the matrix could be either derogatory or defective. The solution of matrix Riccati equation is then calculated from a part of similarity transformation which should reduce the coefficient matrix to the Jordan canonical form.

17. Key Words and Document Analysis. 17a. Descriptors

Ordinary and Partial Differential Equations

Matrix Algebra

Nonlinear and Functional Equations

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement Copies may be obtained from the address in (9) above.

Distribution unlimited.

19. Security Class (This Report) UNCLASSIFIED

20. Security Class (This Page UNCLASSIFIED

21. No. of Pages 38

22. Price NC